TATION PAGE

Form Approvéd
OMB No. 0704-0188

d to average 1 hour per respo se including the time for reviewing instructions, searching ex sting data sources ewing the collection of information Send comments regarding this burden estimate or any other aspect of this den to Washington Headquarters Services, Directorate for formation Operations and Reports, 12 Jefferson ice of Management and Budget, Paperwork Reduction Project (0704-0188), Washington DC 20503

T DATE

3. REPORT TYPE AND DATES COVERED
FINAL 1 Nov 88 - 31 Oct 91

2

4. TITLE AND SUBTITLE
"DEVELOPMENT OF THE ASPECT GRAPH REPRESENTATION FOR USE IN R VISION" (U)

5. FUNDING NUMBERS
61102F
2304/A7

6. AUTHOR(S)
Professor Kevin Bowyer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of South Florida
College of Engineering
4202 Fowler Avenue
Tampa, FL 33620-5350

8. PERFORMING ORGANIZATION REPORT NUMBER
AFOSR-TR 92 0071

9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
Bldg 410
Bolling AFB DC 20332-6448

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFOSR-89-0036

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release;
Distribution Unlimited

12b. DISTRIBUTION CODE
UL

13. ABSTRACT *(Maximum 200 words)*

Over the past few years, a number of researchers have presented algorithms for computing the aspect graph representation for polyhedra and curved-surface objects. However, currently it is computer from the theoretical standpoint of perfect resolution in the viewpoint, the projected image and the object shape. This means that the aspect graph may include details that an observe could never see in practice. This paper reviews a complete implementation of an algorithm to compute the exact aspect graph of solids of revolution under the perspective projection in 3D space. The we explore the notion of introducing scale into the qualitative aspect graph framework, this providing a mechanism for selecting a level of detail that is "large enough" to merit explicit representation. Several alternative interpretations of the scale space aspect graph are examined in response to the results produced for an example object by the implemented system.

14. SUBJECT TERMS

15. NUMBER OF PAGES
23

16. PRICE CODE

| | | | |
|---|---|---|---|
| 17. SECURITY CLASSIFICATION OF REPORT<br>UNCLASSIFIED | 18 SECURITY CLASSIFICATION OF THIS PAGE<br>UNCLASSIFIED | 19. SECURITY CLASSIFICATION OF ABSTRACT<br>UNCLASSIFIED | 20. LIMITATION OF ABSTRACT<br>SAR |

# Final Technical Report for AFOSR 89-0036

Professor Kevin Bowyer

Department of Computer Science and Engineering
University of South Florida
Tampa, Florida 33620
kwb@csee.usf.edu

This is the final technical report for the research grant AFOSR 89-0036, titled *Development of the Aspect Graph Representation for Use in Robot Vision.* This grant covered the three-year period November 1, 1988 through November 30, 1991. The major activities of the first two years of the grant have previously been reported on in the First and Second Annual Technical reports, and so this report only briefly summarizes the activities of the first two years, and concentrates primarily on the activities during the third year of the grant.

The major research results which have come from this work are summarized below.

- Our original algorithm to compute the aspect graph of convex polyhedra was completed [9] and used in a simple recognition system to demonstrate the possible advantages of an aspect graph based recognition system [20, 19].

- The first algorithm for computing the exact perspective projection aspect graph of general polyhedra was developed and implemented [21]. This implementation is being made available to the research community via anonymous ftp.

- The first algorithm for computing the exact perspective projection aspect graph of any class of curved-surface objects was developed and implemented [3, 8, 13]. The particular class of objects addressed in this work was solids of revolution described as right, circular, straight, homogeneous generalized cylinders. The implementation of this algorithm is also being made available to the research community via anonymous ftp.

- The aspect graph concept was generalized from simple rigid objects to objects composed of rigid parts which may have articulated connections between them– "articulated assemblies" [7, 17, 11]. Two different representations for this generalized aspect graph were described, and algorithms were outlined for computing these representations.

- The aspect graph concept was generalized from the ideal assumptions of perfect resolution in viewpoint space, image space, and object shape to finite-scale approximations [10, 1]. This initial "scale space aspect graph" work is our most recent result in the aspect graph area, and potentially opens up a whole new line of research in making the aspect graph better suited for practical use.

- Working with Professor Charles Dyer at the University of Wisconsin, a paper was prepared which provides a tutorial introduction to the aspect graph concept and a survey of recent results [6]. An updated version of this paper has recently been solicited as an invited paper to the 1992 Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS).

- A panel was arranged at the 1991 IEEE Workshop on Directions in Automated CAD-Based Vision, on the theme "Why aspect graphs are not (yet) practical" [12] This panel generated a great deal of discussion, and the updated written comments of the panel will appear as a report in an upcoming special issue of *CVGIP: Image Understanding.*

- In addition to our work in the area of aspect graph algorithms, we have developed a project to investigate the "form and function" paradigm for object recognition. Under this paradigm, the vision system initially has no explicit geometric or structural model for any particular object. Object recognition is performed by reasoning about an object shape to determine the function that it could serve. Our first system implementation to demonstrate this concept used a function-based model for the single object category "chair" [5, 14, 16]. We have just recently completed evaluation of an expanded system which deals with a collection of five separate object categories under the superordinate category furniture [2]. Several additional extensions of this work are currently in progress.

Two appendices have been included with this report in order to provide greater technical detail. The first appendix is a preprint of the paper "Applying the scale space concept to perspective projection aspect graphs," which will appear in the book titled *Selected Papers of the 7-th Scandinavian Conference on Image Analysis.* The second appendix is a reprint of the paper "Achieving generalized object recognition through reasoning about association of function to structure," which has recently appeared in *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

A list of the most important publications resulting from this research begins on the following page.

Eight students have completed Master's theses related to this project, and three students have completed Ph.D. dissertations related to this project. The three Ph.D. students are Louise Stark, John Stewman and David Eggert. Each of the three Ph.D. students was (at different times) partly supported by this grant.

# References

[1] Eggert, D.W., Bowyer, K.W., Dyer, C.R., Christensen, H.I. and Goldgof, D.B. The scale space aspect graph, submitted for publication.

[2] Stark, L. and Bowyer, K. Indexing function-based categories for generic object recognition, submitted for publication.

[3] Eggert, D. and Bowyer, K.W. Computing the perspective projection aspect graph of solids of revolution, to appear in *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

[4] Stark, L., Hall, L.O. and Bowyer, K.W. An investigation of methods of combining functional evidence for 3-D object recognition, to appear in *Knowledge Engineering Review.*

[5] Stark, L. and Bowyer, K.W. Achieving generalized object recognition through reasoning about association of function to structure, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13** (10), (October 1991) 1097-1104; (special issue on interpretation of 3-D scenes, edited by D. Huttenlocher and W.E.L. Grimson).

[6] Bowyer, K.W. and Dyer, C.R. Aspect graphs: an introduction and survey of recent results, *International Journal of Imaging Systems and Technologies* **2**, 315-328 (1990); special issue on computer vision, edited by T.S. Huang.

[7] Sallam, M.Y. and Bowyer, K.W. Generalizing the aspect graph concept to include articulated assemblies, *Pattern Recognition Letters* **12** (3), 171-176 (March 1991).

[8] Eggert, D., and Bowyer, K.W. Computing the orthographic projection aspect graph for solids of revolution, *Pattern Recognition Letters* **11** (11), 751-763 (November 1990).

[9] Stewman, J. and Bowyer, K.W. Direct construction of perspective projection aspect graphs for planar-face convex objects, *Computer Vision, Graphics and Image Processing* **51** (1), 20-37 (July 1990).

[10] Eggert, D.W., Bowyer, K.W., Dyer, C.R., Christensen, H.I. and Goldgof, D.B. Applying the scale space concept to perspective projection aspect graphs, to appear in *Selected Papers of the 7-th Scandinavian Conference on Image Analysis*, P. Johanasen and S. Olsen, editors, World Scientific Publishing, 1992.

[11] Eggert, D., Sallam, M. and Bowyer, K.W. Generalizations of the aspect graph for objects with moving parts, to appear in *Advances in Artificial Intelligence Research: Volume 3*, JAI Press, 1992.

[12] O. Faugeras, J. Mundy, N. Ahuja, C. Dyer, A. Pentland, R. Jain, K. Ikeuchi and Bowyer, K. Panel theme: why aspect graphs are not (yet) practical for computer vision, *IEEE Workshop on Directions in Automated CAD-Based Vision*, Maui, Hawaii (June 1991), 98-104.

[13] Eggert, D. and Bowyer, K.W. Perspective projection aspect graphs of solids of revolution: an implementation, *IEEE Workshop on Directions in Automated CAD-Based Vision*, Maui, Hawaii (June 1991), 44-53.

[14] Stark, L., and Bowyer, K.W. Generic recognition through qualitative reasoning about 3-D shape and object function, *Computer Vision and Pattern Recognition (CVPR '91)*, Maui, Hawaii (June 1991), 251-256.

[15] Wilkins, B., Goldgof, D. and Bowyer, K.W. Toward computing the aspect graph of deformable generalized cylinders, *SPIE #1468: Applications of Artificial Intelligence IX*, Orlando, Florida (April 1991), 662-673.

[16] Stark, L. and Bowyer, K.W. Achieving generalized object recognition through reasoning about association of function to structure, *AAAI-90 Workshop on Qualitative Vision*, Boston, Massachusetts (July 1990), 137-141.

[17] Sallam, M.Y., Stewman, J.H. and Bowyer, K.W. Computing the visual potential of an articulated assembly of parts, *Third International Conference on Computer Vision (ICCV '90)*, Osaka, Japan (December 1990), 636-643.

[18] Eggert, D., and Bowyer, K.W. Computing the orthographic projection aspect graph for solids of revolution, *IEEE Workshop on Interpretation of 3D Scenes*, Austin, Texas (November 1989), 102-108.

[19] Bowyer, K.W., Eggert, D., Stewman, J., and Stark, L. Developing the aspect graph representation for use in image understanding, *DARPA Image Understanding Workshop*, Palo Alto, California (May 1989), 831-849.

[20] Stark, L., Eggert, D., and Bowyer, K.W. Aspect graphs and non-linear optimization in 3-D object recognition, *Second International Conference on Computer Vision (ICCV '88)*, Tarpon Springs, Florida (December 1988), 501-507.

[21] Stewman, J., and Bowyer, K.W. Creating the perspective projection aspect graph of polyhedral objects, *Second International Conference on Computer Vision (ICCV '88)*, Tarpon Springs, Florida (December 1988), 494-500.

# APPLYING THE SCALE SPACE CONCEPT TO PERSPECTIVE PROJECTION ASPECT GRAPHS[1]

David W. Eggert[*] Kevin W. Bowyer[*]
Charles R. Dyer[†] Henrik I. Christensen[‡] Dmitry B. Goldgof[*]

[*] Department of Computer Science and Engineering
University of South Florida
Tampa, Florida 33620 U.S.A.
eggertd, kwb or goldgof@csee.usf.edu

[†] Department of Computer Science
University of Wisconsin
Madison, Wisconsin 53706 U.S.A.
dyer@cs.wisc.edu

[‡] Institute of Electronic Systems
Aalborg University
DK-9220 Aalborg East, Denmark
hic@vision.auc.dk

## Abstract

Over the past few years, a number of researchers have presented algorithms for computing the aspect graph representation for polyhedra and curved-surface objects. However, currently it is computed from the theoretical standpoint of perfect resolution in the viewpoint, the projected image and the object shape. This means that the aspect graph may include details that an observer could never see in practice. This paper reviews a complete implementation of an algorithm to compute the exact aspect graph of solids of revolution under perspective projection in 3-D space. Then we explore the notion of introducing scale into the qualitative aspect graph framework, thus providing a mechanism for selecting a level of detail that is "large enough" to merit explicit representation. Several alternative interpretations of the *scale space aspect graph* are examined in response to the results produced for an example object by the implemented system.

## 1 Introduction

Viewer-centered representations are quite useful in the recognition of objects in a 2-D intensity image [5]. One such representation is the *aspect graph* [19], which is defined as a graph structure in which (1) there is a node for each *general view* of the object as seen from some maximal, connected cell of viewpoint space, and (2) there is an arc for each ***visual event*** (of ***accidental view***) that occurs for a transition across a boundary between neighboring cells. A general viewpoint is defined as one from which an infinitesimal movement in each possible direction in viewpoint space results in a view that is equivalent to the original. In contrast, an accidental viewpoint is one for which there is at least one direction in which an infinitesimal movement results in a view that is different from the original. Under this definition the aspect graph is complete, in that it provides an enumeration of the fundamentally different views of an object, yet is minimal in size since the cells of general viewpoint are disjoint.

The various algorithms that have been developed to date may be classified using three properties; the domain of objects, the view representation and the model of viewpoint space. The domain of objects has evolved from polygons [15], to polyhedra [14, 25,

[1] This work was supported at the University of South Florida by Air Force Office of Scientific Research grant AFOSR-89-0036, National Science Foundation grant IRI-8817776, and a Florida High Technology and Industry Council committee on Computer Integrated Engineering and Manufacturing grant.

29, 31, 33, 34], to solids of revolution [9, 10, 20], to piecewise-smooth objects [7, 26, 27, 30], to articulated assemblies [28]. Almost without exception, a view of the object is represented using a qualitative description of the line drawing, such as the *image structure graph (ISG)* [22]. The actual labeling of *contours* and *junctions* varies slightly among researchers. Distinctions between general and accidental views are usually based on isomorphism of the ISG. Lastly, two viewpoint space models are commonly used. The first is the 2-D *viewing sphere*, on which each point defines a viewing direction for *orthographic projection*. The other is 3-D space, in which each point is the focal point for a *perspective projection*. (For greater detail on these algorithms, see [4, 11].)

Recently the practical utility of the resulting aspect graphs has been questioned. At the 1991 *IEEE Workshop on Directions in Automated CAD-Based Vision* a panel discussion on the theme "Why aspect graphs are not (yet) practical for computer vision" was held [13]. One issue raised by the panel is that aspect graph research has not included any notion of *scale*. In order to address this issue we have developed the concept of the *scale space aspect graph*. This representation is seen as a method of countering the ideal assumptions made concerning perfect resolution in viewpoint, the projected image and object shape that can lead to practical difficulties.

In section two we review a particular aspect graph creation algorithm [10, 11] and examine the resulting representation for a flower vase object. In section three we define the scale space aspect graph and its properties. Section four details three different interpretations of the scale parameter that deal with the above ideal assumptions. Conclusions and directions for future research are discussed in section five.

## 2 The Aspect Graph of a Solid of Revolution

In this section an overview is given of an algorithm that constructs an aspect graph and its implementation [10, 11]. The domain of objects consists of those *solids of revolution* defined by using a Generalized Cylinder model. The *sweeping rule*, or profile curve, is assumed to be piece-wise continuous, single-valued, and continuously differentiable. Each piece of the sweeping rule is described by an arbitrary degree, positive-valued polynomial function of the length along the object axis. Furthermore, only opaque, matte solids without surface markings, specularities or shadows are considered. Views of the object are represented using an *image structure graph* to be described shortly. Viewpoint space is defined as all of 3-D space, excluding the object volume, and the perspective projection viewing model is used.

### 2.1 Algorithm overview

The algorithm to compute the perspective projection aspect graph of solids of revolution can be summarized using the following steps:

**1. Determine the "lines" (or "contours") that may interact in a view.** Contours are of two types, *edges* and *limbs*. Edges are the convex-shaped (with respect to the object axis) projections of surface tangent discontinuities at an object end or between pieces. Limbs (occluding contours) are the projections of points on the object surface (contour generators) where a line of sight is tangent to the the object surface. Limbs are convex or concave shaped depending on whether the portion of the surface for which they are the projection is elliptic or hyperbolic, respectively. In order to keep track of each type, the object is subdivided into elliptic and hyperbolic regions, one for each limb. Also, hyperbolic regions are further divided at positions where a *cusp* first occurs (a single limb splits into two pieces, one occluding the other and terminating at a point), one for each segment. A view is described using a labeled line drawing known as an image structure graph (ISG) [9, 22]. Arcs in the ISG are labeled according to contour and projected region type, while nodes, are labeled according to quantity, connectivity, and type of contours intersecting at the point (See Figure 3 for examples). Two views are considered equivalent if and only if their corresponding ISGs are isomorphic.

**2. Determine the visual event surfaces.** The types of surfaces in 3-D space that can be generated by accidental alignments of features are limited. Clearly, the surfaces must be ruled, as they are composed of families of lines of sight. Also, due to the rotationally symmetry of the object, views from points along a circle centered about and perpendicular to its axis will be the same. Thus, even the event surfaces must be rotationally symmetric about the object axis. Only four such surfaces exist: a plane perpendicular to the axis, a cylinder, a circular cone, and a hyperboloid of one sheet. The visual events that generate these four surfaces fall into three general categories:

*Individual Events* - Since limbs are viewpoint dependent, each region for which they are the projection has a defined range of potential visibility. This range is bounded by surfaces (no planes) that are tangent to the object surface at the ends of the region.

*Pair Events* - Limbs and edges taken in pairs may interact. The most common interaction is occlusion. Two surfaces are generated, one that marks first contact between contours, and one for final contact (usually one contour is completely hidden at this point) as one moves toward the object. Planes and hyperboloids are generated due to initial contact of the contours at two symmetric points in the image. Final contour contact in the image is marked by cones and cylinders (at one point) and hyperboloids (at two symmetric points). Nonocclusion interactions involve the formation / disappearance of various junctions when contour generators (creases) from neighboring regions make contact / split apart at a point on the object surface. These events generate surfaces (again no planes) that are tangent to the object surface at the point of contact. In addition, the planes containing the ends of object pieces mark the transformation of junction type between edge and neighboring limbs.

*Triplet Events* - Three contours can appear to coincide at symmetric points in the image, the event surface being a hyperboloid. Before and after this coincidence only two of the three pair occlusion intersections are visible (different ones for each). In actuality, this event marks the first contact of occlusion between the outermost pair of contours.

The accidental alignments that define a visual event impose constraints on its surface parameters that translate into a system of polynomial equations. The systems for nonocclusion and certain occlusion events can be solved directly. However, numerical searches are necessary to solve the systems for most occlusion events. A geometric technique is used to structure the searches. Since the the solution surface form is known, a subset of the constraints will directly generate parameters of a potential surface, given the value of one parameter. The remaining constraints yield an error measure for this surface in a binary search for the value of the chosen parameter.

**3. Parcellate viewpoint space.** Because of rotational symmetry, a subdivision of 3-D space is sufficiently described using the parcellation of a half-plane containing the object axis. In the implementation (described shortly), it is assumed that the object axis coincides with the $Z$ axis, while the upper half ($X > 0$) of the $XZ$ plane is chosen for the parcellation. The curves of intersection between the event surfaces and this space, relative to the $Z$ axis, are: a perpendicular line, a parallel line, two lines of opposite slope meeting at a point on the $Z$ axis and one half of a hyperbola.

Each visual event surface has some meaningful range. For instance, the portion of an occlusion event surface between the interacting regions is not important. Also, those portions extending out from the interacting points are unimportant after penetrating the object surface (if ever), due to global occlusion. Since the event curves in the $XZ$ plane are single-valued with respect to $Z$ (excepting perpendicular lines), a modified plane sweep algorithm is used to organize the incremental construction of the parcellation. This data structure is composed of cells (regions in the plane) defined by a set of bounding curves, which in turn are defined by the intersection points terminating them.

4. **Create the aspect graph and representative views.** The aspect graph (which has a 1-to-1 correspondence in structure to the parcellation) is constructed incrementally during a traversal of the parcellation. At the same time, it is also possible to determine the representative view of each aspect. From the most distant side view of the object, every limb and edge is visible and connected together in a predictable manner. If one then crosses each event surface by moving towards the object, the change in view is either a relabeling or restructuring of visible entities, or limbs may disappear. By using a depth-first traversal through the parcellation, begun at the side view cell, it is possible to incrementally generate ISGs of the views according to the visual events without resorting to hidden-surface calculations.

## 2.2 The implementation

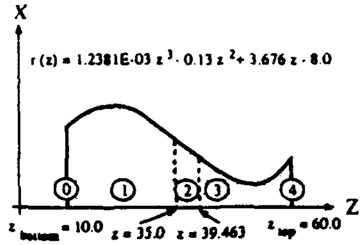
The implementation of the algorithm (approximately 38,000 lines of C)[2] includes a visualization package (using X-windows) for observing the creation process, as well as viewing the object, its aspects, and the parcellation. Input to the system consists of an object definition file containing the polynomial equations and ranges of the piece-wise profile curve. The output file contains information characterizing each aspect's ISG, as well as sufficient data to reconstruct the aspect graph and its underlying parcellation.

The two main difficulties encountered during the system's development were numerical precision and solving systems of polynomial equations. An extended-precision package[3] was incorporated to deal with operations on polynomials, since double precision arithmetic was insufficient for accurate evaluation of "large" (say, seven or eight) degree polynomials. This greatly increased calculation reliability, but at a (great) reduction in speed. The second problem concerned developing numerical searches to solve the polynomial systems.Techniques such as numerical continuation and elimination theory [26] were considered, but reliable results across our database of objects could not be obtained. The eventual geometry-based searches already discussed converged for all test cases, and were more efficient than the general techniques in many cases.

The system has constructed aspect graphs of over 100 different objects, reliably handling those with sweeping rules of at least degree ten. (For results see [11].) The database ranges in complexity from a cylinder (5 aspects, 0 finite-extent and 5 infinite-extent cells) to an object with a degree eleven sweeping rule (829 aspects, 767 finite-extent and 62 infinite-extent cells). Execution times on a SUN Sparc 1+ ranged from approximately ten seconds for the cylinder to 24 hours for the more complex object, while generating output files of size 3.5KB and 962KB, respectively. Because the aspect graph generation is an "off-line" process, and its use an "on-line" process, the system was designed for flexibility and accuracy, rather than speed and minimum output size.

[2] The software is available to interested researchers - contact David Eggert or Kevin Bowyer.

[3] The actual package used is the Arbitrary Precision Math Library developed by Lloyd Zusman, Master Byte Software, Los Gatos, California, U.S.A.



| Number | Region type | Range |
|---|---|---|
| 0 | Lower Edge | $z = 10.0$ |
| 1 | Complete Elliptic | $10.0 < z < 35.0$ |
| 2 | Lower Hyperbolic | $35.0 < z < 39.463$ |
| 3 | Middle Hyperbolic | $39.463 < z < 60.0$ |
| 4 | Upper Edge | $z = 60.0$ |

Figure 1: Definition of flower vase sweeping rule, r (z), and interacting regions.

### 2.3 An example

As an example of the system's performance, we have chosen an object analyzed in previous papers [9, 20] (see Figure 1.a). This object took five minutes to process resulting in a 30KB output file. There are five surface regions which project to contours in the image (see Figure 1.b). Upon calculating the visual events for the object, eleven event surfaces (composed of nineteen meaningful portions) were found: six hyperboloids of one sheet, three circular cones, and two vertical planes. The defining curves in the $XZ$ plane, along with the corresponding events, are listed in Table 1. From this set of curve segments the parcellation of the $XZ$ plane in Figure 2 was calculated. There are a total of 49 aspects, numbered according to the traversal ordering established when forming the aspect graph. Eighteen of these have infinite–extent viewing cells, but only seventeen correspond to general views using orthographic projection [9]. The inconsistency is cell 1, the initial side view. Because of its nonexpanding cross–section, this cell only corresponds to an accidental view (from the equator of the viewing sphere). In Figure 3 views of the object (produced by the system) are drawn for an orbit along its axis. The corresponding ISGs for these aspects are also shown.

## 3 The Scale Space Aspect Graph

Now that some "typical" results for an aspect graph have been presented, we are in a position to comment on the weaknesses of the representation and propose potential improvements. These weaknesses arise from various assumptions that were made. In this paper we do not deal with the explicit assumptions, such as the use of the ISG as a view representation, since these vary among the known algorithms. Instead we focus on problems inherent to the approach, which have perhaps a more fundamental impact on aspect graph usage. These center around the qualitative nature of the representation, i.e., the lack of scale information. Three of these basic assumptions are:

1. **The camera is idealized as a point.** This assumption manifests itself in the fact that each node in the aspect graph represents a view of equal significance. The underlying shape and size of the cell in the parcellation has bearing on its importance. Since a camera does have a finite size, certain views are unlikely to ever be witnessed. For example, notice the several narrow and small cells in the parcellation of Figure 2.

2. **There is infinite resolution in the projected image.** In this case each feature in the ISG is accorded equal significance. This means that a given view may have a feature that is too small to detect from within its cell, and two views may differ by only such a feature and therefore be the same in practical terms. Note the size of some of the hyperbolic limbs ending in cusps in Figure 3. Also, each portion of the line drawing is distinguishable at an infinite distance, a definite departure from reality. This leads to

| Index | Event Curve Equation | Range | Visual Event Description |
|---|---|---|---|
| 1 | $\frac{x^2}{74.214} - \frac{(s-55.983)^2}{17.163} = 1$ | $60 < s < +\infty$ | end joining of contours 3 and 4,<br>begin occlusion of contour 3 by contour 4 |
| 2 | | $-\infty < s < 60$ | end joining of contours 3 and 4,<br>begin occlusion of contour 4 by contour 3 |
| 3 | $\frac{x^2}{40.944} - \frac{(s-48.074)^2}{41.158} = 1$ | $39.463 < s < +\infty$ | begin occlusion of contour 2 by contour 3 |
| 4 | | $-\infty < s < 39.463$ | begin occlusion of contour 3 by contour 2 |
| 5 | $\frac{x^2}{38.103} - \frac{(s-48.744)^2}{39.166} = 1$ | $-\infty < s < 35$ | end occlusion of contour 3 by contour 2,<br>begin occlusion of contour 3 by contour 1 |
| 6 | | $44.519 < s < +\infty$ | end occlusion of contour 2 by contour 3 (contour 2 now hidden),<br>begin occlusion of contour 1 by contour 3 |
| 7 | $\frac{x^2}{35.77} - \frac{(s-48.898)^2}{40.738} = 1$ | $60 < s < +\infty$ | end occlusion of contour 3 by contour 4 (contour 3 now hidden) |
| 8 | $\frac{x^2}{34.818} - \frac{(s-48.976)^2}{38.75} = 1$ | $60 < s < +\infty$ | triplet occlusion interchange,end occlusion of contour 1 by contour 3<br>begin occlusion of contour 1 by contour 4 |
| 9 | | $-\infty < s < 34.346$ | triplet occlusion interchange,end occlusion of contour 4 by contour 3<br>begin occlusion of contour 4 by contour 1 |
| 10 | $\frac{x^2}{31.098} - \frac{(s-50.501)^2}{40.773} = 1$ | $-\infty < s < 31.483$ | end occlusion of contour 3 by contour 1 (contour 3 now hidden) |
| 11 | $x^2 - \frac{(s+1.743)^2}{0.477} = 0$ | $-\infty < s < -1.743$ | end joining of contours 0 and 1 (contour 0 within contour 1) |
| 12 | | $-1.743 < s < 10$ | end joining of contours 0 and 1,contour 1 no longer exists |
| 13 | | $10 < s < +\infty$ | end joining of contours 0 and 1 (contour 0 now hidden) |
| 14 | $x^2 - \frac{(s-51.594)^2}{1.31} = 0$ | $-\infty < s < 35$ | end joining of contours 1 and 2,contour 2 no longer exists |
| 15 | $x^2 - \frac{(s-100.892)^2}{11.612} = 0$ | $100.892 < s < +\infty$ | end occlusion of contour 1 by contour 4 (contour 4 within contour 1) |
| 16 | | $60 < s < 100.892$ | end occlusion of contour 1 by contour 4 (contour 1 now hidden) |
| 17 | | $-\infty < s < 22.5$ | end occlusion of contour 4 by contour 1 (contour 4 now hidden) |
| 18 | $s = 10.0$ | $17 < x < +\infty$ | transform junctions (3-tangent — curvature-L) of contours 0 and 1 |
| 19 | $s = 60.0$ | $12 < x < +\infty$ | transform junctions (3-tangent — curvature-L) of contours 3 and 4 |

Table 1: Definitions of visual event curves for flower vase.

infinite-extent cells, when there should be a finite limit to meaningful viewing distance.

**3. The object shape is known in minute detail.** Visual events are generated through interactions of the various surface portions. Small bumps or indentations may generate several event surfaces, the visual changes of which might be considered insignificant. Also, certain event surfaces might just exist due to a fragile alignment. Thus a small change in the object definition may drastically alter the set of potential events, and the shape of the parcellation. One can imagine that a flower vase with a slightly different shape than that in Figure 1 would have a different number of aspects.

Each of these factors seems to contribute to a representation that is larger in size than is realistic. (For example, the worst-case node complexity is $O(N^4)$ for a solid of revolution defined by an Nth-degree polynomial assuming a 3-D viewpoint space.) By introducing the concept of scale into the representation we hope to reduce this large set of theoretical aspects to a smaller set of the "most important" aspects.

This new representation will be termed the *scale space aspect graph*. In its strictest sense, the phrase "scale space of X" is taken to mean a parameterized family of X in which the detail of features in X is monotonically decreasing with increasing scale. Also, the qualitative features of X at a given scale can be traced back across all lower scales ("causality"). This topic was popularized by Witkin's scale space analysis of a 1-D signal [35]. Since that time the scale space concept has been applied to the curvature of 2-D curves [6, 23], the curvature of 3-D curves [24], the 2-D intensity map [1, 17, 21, 36] and 3-D object shape [18]. In addition, a number of other researchers have described similar "hierarchical" or "multi-resolution" representations, such as pyramids.

In Witkin's original analysis, the qualitative structure of a 1-D signal was given in terms of inflection point locations. The 2-D scale space of a 1-D signal is developed by introducing a second dimension, $\sigma$, that represents the size of a Gaussian kernel used to smooth the original signal. In this parameterized family of signals, a value of $\sigma = 0$

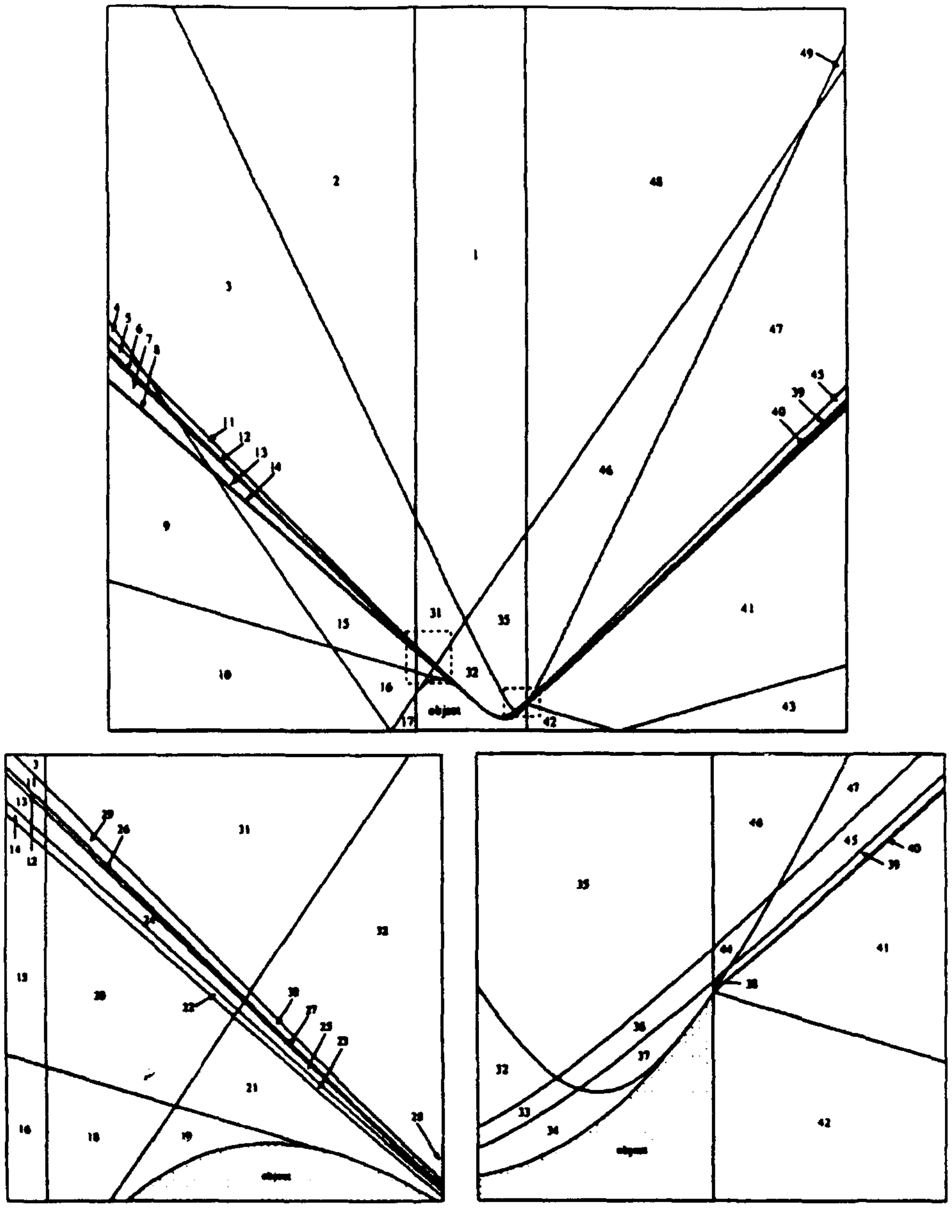


Figure 2: Parcellation of $XZ$ plane using visual event curves described in previous figure.

yields the original, while $\sigma = \infty$ reduces the signal to a flat line. In the scale space, a particular inflection can be traced over increasing values of $\sigma$ until it is eventually annihilated (merged with a neighboring inflection). In keeping with the monotonicity requirement, inflection points can only be annihilated as $\sigma$ increases, never generated. Thus the scale at which an inflection ceases to exist is a measure of its strength.

By now, the definition of a *scale space aspect graph*, at least at a high level, should be apparent. Since the aspect graph is nothing more than a qualitative description of the underlying structure of the parcellation of viewpoint space, it is appropriate to consider

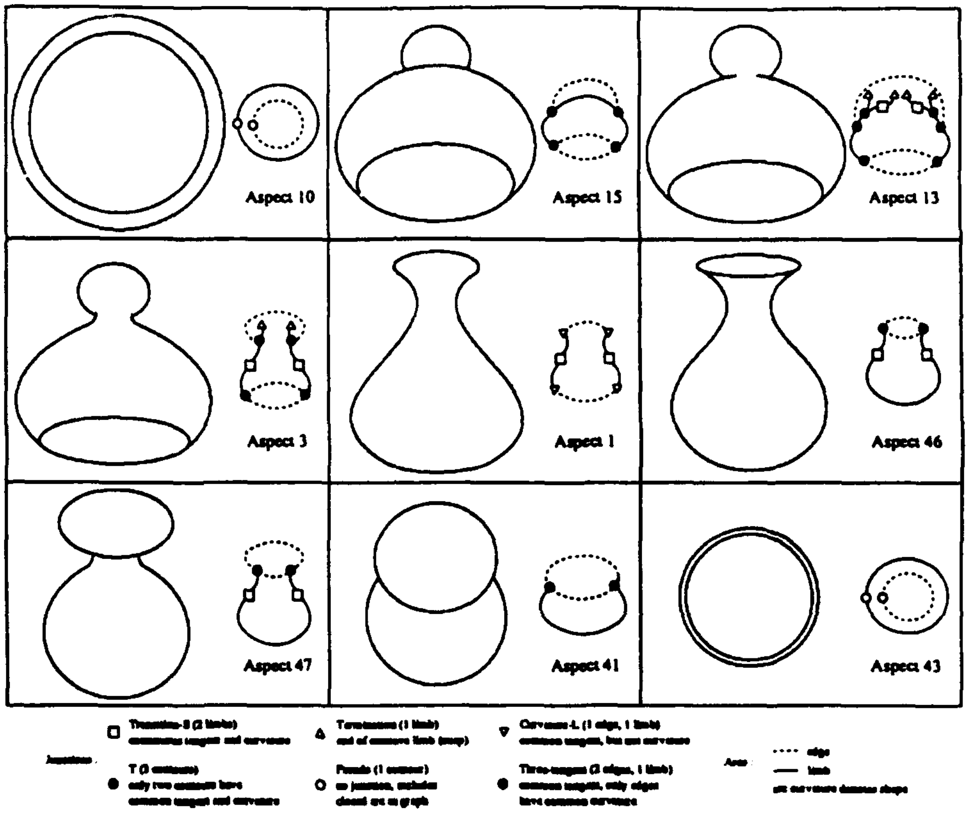
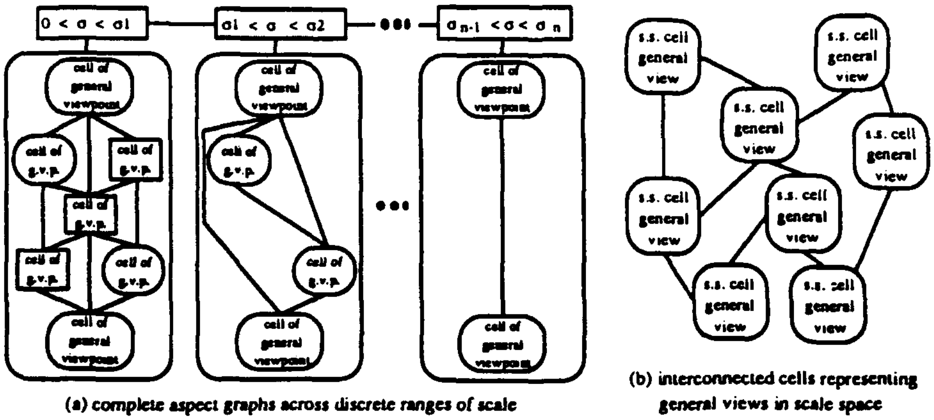

Aspect 10
Aspect 15
Aspect 13
Aspect 3
Aspect 1
Aspect 46
Aspect 47
Aspect 41
Aspect 43
Junctions:
Transition-S (2 limbs) continuous tangent and curvature
Termination (1 limb) end of concave limb (cusp)
Curvature-L (1 edge, 1 limb) common tangent, but not curvature
T (3 contours) only two contours have common tangent and curvature
Pseudo (1 contour) no junction, includes closed arc in graph
Three-tangent (2 edges, 1 limb) common tangent, only edges have common curvature
Arcs:
edge
limb
arc curvature denotes shape

Figure 3: Views and ISGs of aspects along orbit about object at radius of 75 units.

a parameterized family of these parcellations as the basis for the scale space. This scale space is defined as a 4-D space $(x, y, z, \sigma)$ parameterized by viewpoint location and scale value. Each visual event surface is now a function of both viewpoint and scale. Thus, at $\sigma = 0$ the parcellation of the viewpoint space, and so also the aspect graph, is exactly as computed by some known aspect graph algorithm. As $\sigma$ increases, the parcellation of viewpoint space should deform in a way such that at certain discrete values of scale the aspect graph becomes simpler (has fewer nodes).

There are (at least) two alternative representations of the qualitative structure of scale space as shown in Figure 4. The first, an explicit sequence of aspect graphs over consecutive ranges of $\sigma$ in which its structure is constant, is perhaps simpler conceptually, but potentially has a great deal of redundancy in the multiple instances of the aspect graph. This form bears resemblance to the *visual potential* of Sallam *et al.* [28]. In their representation separate instances of the aspect graph are recorded for varying articulation parameter values of an object. Here scale can be thought of similarly.

The second, a more compact representation, is directly analogous to the typical form of the aspect graph. Each node represents a "volume" of the scale space for which the same general view exists. Each arc again represents a visual event, but the underlying boundary is now parameterized by the scale dimension. This form corresponds most closely to the *asp* of Plantinga and Dyer [25]. In their representation the aspect graph was formed as the projection of certain higher-dimensional "ˈolumes", represent-



(a) complete aspect graphs across discrete ranges of scale

(b) interconnected cells representing general views in scale space

Figure 4: Conceptual Depictions of the Scale Space Aspect Graph.

ing particular feature configurations, into the viewpoint space. This is essentially the conversion process used to elicit whatever information is necessary for a particular scale. Other representations, such as extensions of the *interval tree* concept [21, 35], may exist depending on the interpretation of the scale parameter, the topic of the next section.

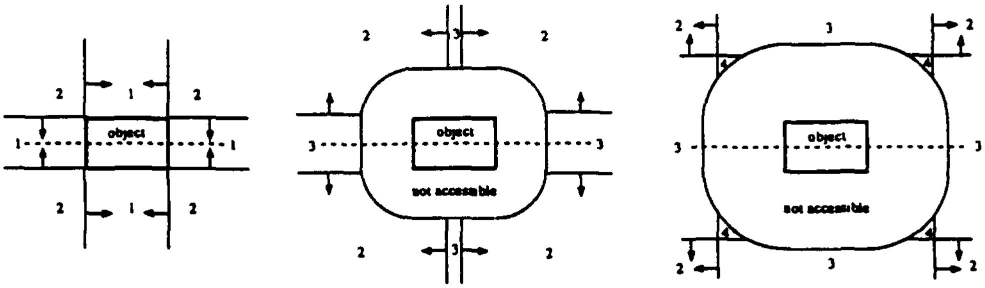
## 4 Interpretations of Scale

We must now speculate on how one might use a single scale parameter (or possibly more) to create a family of parcellations of the viewpoint space. There is no one unique possibility. Previous scale space representations have been applied to 1-D, 2-D and 3-D intensity functions by interpreting the scale parameter in terms of the solution to the diffusion equation [17] (or more specifically, as the variance of a Gaussian kernel used to blur the function). It has been proven that only under this interpretation will the qualitative features of the function disappear and not be created as the scale value is increased [17]. However, since the entities on which the aspect graph concept is based (such as visual events, projected line drawings, and 3-D shape) are not intensity functions, it is hard to define what one means by "blurring" the parcellation of viewpoint space. Therefore the requirement that the quantity of features monotonically decrease in size may have to be relaxed. We now examine the three problems addressed earlier in search of interpretations of "blurring" the parcellation.

### 4.1 Scale of viewer relative to cell of viewpoint space

One interpretation is to examine the relative sizes of cells in viewpoint space with respect to a finite-sized observer. In the past researchers have considered the probability of certain views based on relative cell volumes [2, 11, 16, 33, 34]. However, we propose a more extensive relation of viewer and cell, that corresponds more intuitively to blurring the existing parcellation. In this we relax the assumption that the viewer is idealized as a point. Instead, a finite-sized sphere, the radius of which is a function of scale, will model the area of space in which light rays may be gathered and directed onto the image. (Imagine rotating the circular lens of a camera about the focal point to sweep out the volume of a sphere.) Any light impinging upon this sphere contributes to the composite image, as observable features from each point in the sphere are merged.

This interpretation can also be explained in terms of changes in the parcellation as follows. For a given size sphere there will still be a range of viewpoints in a typical cell



arrows indicate direction of movement of event surfaces, numbers indicate quantity of surfaces visible from region

(a) initial structure of parcellation (b) parcellation after overlap of 2-face areas (c) parcellation after overlap of 3-face areas

Figure 5: Changes to parcellation of cylinder based on expanding sphere radius.

for which the sphere is fully contained within the cell. For those viewpoints from which the sphere pierces the cell boundary, a composite view exists made up of those views from the cell, the accidental boundary, and the neighboring cell. In certain cases this view will be equivalent to that of one of the cells. For example, consider a visual event surface (curve 18 between cells 1 and 2 in Figure 2) that marks the occlusion boundary for a face of the object. The composite view is the same as the one in which the face is visible. Thus the size of the multiple–face cell is impinging upon the area where the face is hidden, by a layer of thickness equal to the viewing sphere radius. In other cases the accidental view is really the composite view itself. For example, consider the event surface (curve 8 between cells 39 and 40 in Figure 2) representing a triple occlusion point in the image. In the ideal case this alignment is only visible from the surface, but for a given size sphere, superimposing this view with those of the neighboring cells merely increases the apparent size of the triple point, as the nearby T junctions all merge together. Therefore, in this instance, the formerly accidental view can be seen from a volume of space and is now a "stable" view.

So we can model the changes to the parcellation by extending the visual event surface positions by the radius of the current viewer sphere in one or two directions depending on its type. If extensions occur in both directions a new general view is added to the aspect graph. In addition to event surface extensions, the extent of viewing space is reduced by a layer extended out from the object surface, since the camera can only get within a certain distance of the object now. As scale (sphere radius) changes certain cells are eliminated from the parcellation, while others come into existence. For those cells being shrunk on all sides, they will cease to exist at a scale that corresponds to the maximal size sphere at a point on the skeleton of the original cell produced by a medial axis transform [3]. Notice that there may be several local maxima along the skeleton, meaning the cell may exist as separate portions before being completely eroded. At the time these cells cease to exist other cells are created by the overlap region of the expanding cells. In these areas a composition of the two views is again formed. It is also possible for these types of cells to be formed from expanding overlap regions.

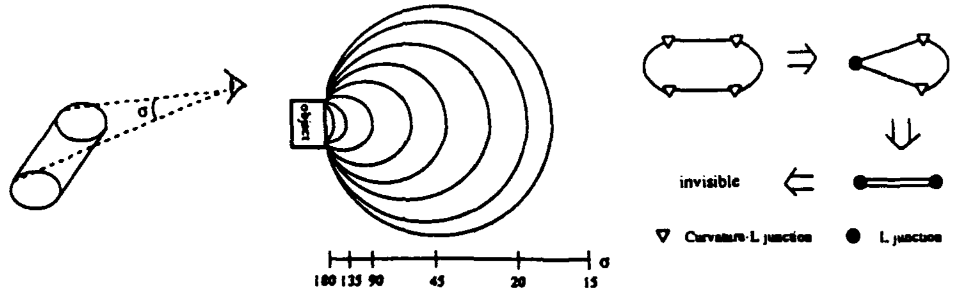
Some of these occurrences are shown in Figure 5, which shows three stages in the development of the parcellation of a cylinder. In the 2–D parcellation the viewer sphere becomes a circle. In the beginning each event surface is extended from a two surface view's cell into that of a one surface view. After time the one surface views are eliminated and replaced by the overlap area in which three surfaces are seen at once. Finally, the overlap of these regions (in which the entire object can potentially be seen) emerges from the region about the object that the camera cannot enter. If this final frame is continued to where the scale is infinity, then there will be no viewing area left in which the camera fits. The importance of the various aspects could be ranked according to the scale at which the cell disappears. But in this case those infinite ranging cells would be ranked equivalent. Perhaps a more accurate ranking is according to the "volume" of the scale space cell composed of the shape of the aspect's cell over all scales. In addition, one may not want to examine the entire scale range up to infinity, as this is somewhat unrealistic. In the next section we see one alternative to this infinite cell interpretation.

### 4.2 Scale of features in the projected image

The features in the image could be analyzed in at least two ways, according to their projected nature in the image intensity function, or in terms of their apparent size as a function of viewpoint position. In terms of analyzing the image intensity function there are also a couple of possibilities. Given assumptions about object surface (say matte in texture) and light source placement (a point light source coincident with the viewpoint) an image intensity function can be constructed. Such a function can be subjected to Gaussian smoothing as a function of scale, and the resulting features analyzed. In terms of the projected line drawing one would keep track of the edges detected in the smoothed image that are above a given magnitude threshold. Thus "weaker" edges would disappear first, and the strength of an edge ranks its importance. An alternative is to describe the image according to the surface topology of the intensity function, e.g., the "hills and dales" representation used by Koenderink [17]. He has studied the changes that occur for a given image under Gaussian smoothing, while others are beginning to explore the types of visual events that exist for such a representation [32]. One difficulty with this approach is that current theory that predicts changes in the ISG is not applicable, since the image is very closely tied to the viewpoint. Therefore we now concentrate on using scale as a measure of the size of features in the projected line drawing.

In this approach the scale dimension affects the resolution of our image, and thus our ability to detect a feature. Also, this method implicitly accounts for size effects due to viewing distance. Some of these ideas are similar to those used by researchers determining visibility constraints for automatic sensor placement [8]. First one must determine which features should be concentrated upon. In order to be measured, a feature must have some spatial extent in the image. This means that a junction, which occurs at a single point, should not be a feature. Alternatively, edges (limbs) and object faces (portions of surface patches) generally have measurable extent in a view. So, how does one quantify the size of a feature? It is not sufficient to measure the length of an edge or area of a face on the object. It is the projection of these features that matters.

The first solution which comes to mind is to measure the dimensions of features in an image coordinate system, the resolution of which is based on our scale parameter. The length along a projected edge, the perimeter or area of a face, or possibly the radius of the sphere that circumscribes the feature would be quantified in terms of a number of pixels. Unfortunately, this approach requires a more detailed camera model; the focal distance, the image plane size (field of view), the particular viewing direction and the viewing position must be known. While such a sophisticated model would be more realistic, it is too complex to consider as a first step. An alternative measurement is the angle of visual arc $\sigma$, or field of view, occupied by the feature. Given the assumption of a "360° eye" used by many aspect graph researchers, every feature's size can be described by one parameter value in the range $0° - 360°$. Exactly how this value is measured depends



(a) angle of visual arc spanned by limb (b) progression of visibility limit boundary of limb toward object for increasing scale (c) progression of side views for increasing scale

Figure 6: Various features of image resolution interpretation of scale.

on the feature. For a straight edge, the distance between its projected endpoints will span a particular visual arc, as shown in Figure 6.a. For a curve, the maximum distance between any two projected points along its length indicates the visual extent. For a face, one must consider the maximum inscribable circle for the projected outline.

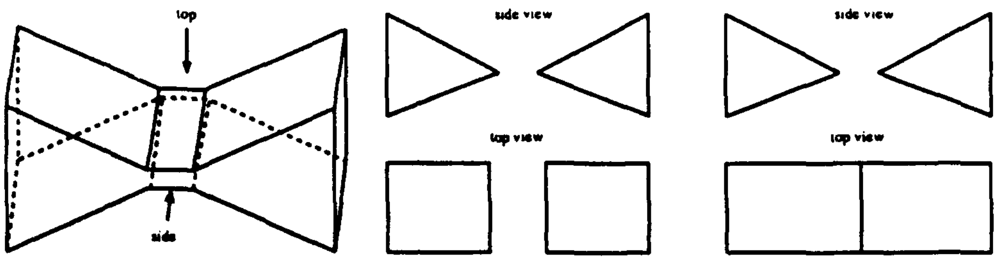
So how is the above interpretation used? It should be obvious that image resolution can be defined in terms of degree of visual arc. Pixel size in the image directly corresponds to the minimum visual arc necessary to distinguish a feature. At a value of 0° the camera has infinite resolution. At a value of 360° there is only a single pixel in the image and everything projects to it. For a given scale, any feature mapping to a size smaller than one pixel is considered as not observable. More exactly, the image resolution has a direct effect on the shape of the visual event boundaries.

In reconsidering the cylinder example, a given feature such as the limb appears at the critical size for a set of viewpoints (typically circular in nature), see Figure 6.b, which varies as a function of the scale parameter. Within the bounds of this set the feature is visible, outside it is not. To see how this affects the view of a face, consider the side view of a cylinder from near one of the ends. As one increases the visual arc necessary to distinguish a feature, the form of the view will follow that indicated in Figure 6.c. First the nearer edge segment will appear as a point, and then the other, since the greatest apparent width for the cylinder is directly under the viewpoint. Lastly even this is too small, and the entire face falls below the resolution of a pixel. This view sequence also occurs as one backs away from the object (agreeing with our intuition). Each feature will pass from sight as the viewpoint moves outside the range from which it is visible.

To construct a scale space aspect graph under this interpretation, one must develop equations for the new event surfaces as a function of arc angle. One then examines how the parcellation structure, which is of a finite size for any nonzero scale value, changes as it goes from the ideal ($\sigma = 0°$) to collapsing about the object. These changes include rearranging the order of intersections, changing the overlap of two viewing regions and noting the end of existence of certain surfaces as features are no longer visible. Such an analysis has been performed for the case of a nonconvex polygon in a plane [12]. Again, aspect importance should be ranked according to cell volume in scale space.

### 4.3 Scale of features of object shape

In this section the effects of altering object shape according to a scale parameter are discussed. One would hope these effects correspond to the loss of detail noticed while moving away from the object. Intuitively, one wants to smooth the object surface until



(a) "bowtie" shaped object (b) views after "dynamic shape" smoothing (c) views after "range" smoothing

Figure 7: The effects of object smoothing on views of bowtie object.

a somewhat featureless blob is achieved, examining the parcellation along the way. The question is how to do the smoothing. For a solid of revolution, one might think about smoothing only the profile curve, which would eventually achieve a cylindrical shape. But this still leaves sharp edges one would not expect to exist on a "smoothed" object. A more general technique proposed recently is the "dynamic shape" concept [18]. This is a form of 3-D volumetric blurring in which the surface is marked as the level set of the resulting distribution. For instance, if the "bowtie" object in Figure 7.a were to be subjected to this process, for a given level of smoothing the central portion of the object would cease to exist and views of it would appear as shown in Figure 7.b. While the view from the side might seem a logical consequence of smoothing the object, the view from the top does not. One would most likely expect to see the views in Figure 7.c. This is because the volumetric smoothing works upon *solid* shape, while that which is observable is *surface* shape. Furthermore this surface shape is relative to the position of the viewpoint, as an inch deep hole seems much larger up close than far away.

Thus we propose a different smoothing approach, which is basically to smooth the range image generated for a particular viewpoint. This smoothing is done in the direction perpendicular to the viewing direction, in a manner similar to smoothing the image intensity function. Given such an approach the views in Figure 7.c could now be expected. Also, the visual event surfaces generated by different portions of the object will now be highly dependent on viewpoint position for their existence. Two portions of the object that interact from one vantage may not have the same relative shape and position to do so from another. Eventually the interaction will no longer occur for any viewpoints as the smoothing increases. Taking this to the extreme the object shape should tend toward an ovoid with no visual event surfaces. Again one should keep track of the parcellation structure as the amount of smoothing is increased until the eventual featureless state is reached.

## 5 Summary and Conclusions

In this paper we have examined the practical utility of the aspect graph representation. Based on the results produced by an actual implemented system for solids of revolution three general weaknesses were noticed: (1) the use of a point observer leads to cells of negligible size, (2) the use of an infinite resolution image plane leads to an imbalance in feature importance, as well as unrealistic infinite-extent cells, and (3) small details of the object may generate many insignificant visual event surfaces. Then the notion of the scale space aspect graph was proposed to evaluate the importance of the views as a particular element of the viewing process was adjusted. These included modeling the viewer as a finite-sized sphere, varying the image resolution, and smoothing out object surface detail. While each of these approaches seems to incorporate a bit more of reality into the representation, each alone has drawbacks. For instance, increasing the viewer's size to infinity seems extreme, and infinite-extent cells continue to exist until that point. By incorporating image resolution the finite nature of cells is achieved, but there are still many small cells and extraneous visual events. Finally, parcellations based on object smoothing suffer deficiencies similar to those for viewer size, and reducing the object to a blob may also be extreme. Thus while we have made important strides in analyzing each phenomenon individually, it is now equally important to study their interrelations. By considering the visual changes as a whole, we may be able to perceive a unifying interpretation. This will most likely lead to a comprehensive model requiring the use of multiple scale parameters, or perhaps other alternatives not discussed here.

## References

[1] Babaud, J., Witkin, A.P., Baudin, M. and Duda, R.O. 1986. "Uniqueness of the Gaussian kernel for scale-space filtering", *IEEE Trans. PAMI*, **8**, 26-33, 1986.

[2] Ben-Arie, J. "Probabilistic models of observed features and aspects with application to weighted aspect graphs", *Pattern Recognition Letters*, **11**, 421-427, 1990.

[3] Blum, H. "Biological shape and visual science (Part I)", *J. Theoretical Biology*, **38**, 205-287, 1973.

[4] Bowyer, K.W. and Dyer, C.R. "Aspect graphs: An introduction and survey of recent results", *Int. J. Imaging Systems and Technology*, **2**, 315-328, 1990.

[5] Bowyer, K., Eggert, D., Stewman, J. and Stark, L. "Developing the aspect graph representation for use in image understanding", *DARPA IUW*, 831-849, 1989.

[6] Clark, J.J. "Singularity theory and phantom edges in scale space", *IEEE Trans. PAMI*, **10**, 720-727, 1987.

[7] Chen, S. and Freeman, H. "On the characteristic views of quadric-surfaced solids", *IEEE Work. Directions in Automated CAD-Based Vision*, 34-43, 1991.

[8] Cowan, C.K. "Automatic camera and light-source placement using CAD models", *IEEE Work. Directions in Automated CAD-Based Vision*, 22-31, 1991.

[9] Eggert, D. and Bowyer, K. "Computing the Orthographic Projection Aspect Graph for Solids of Revolution", *Pattern Recognition Letters*, **11**, 751-763, 1990.

[10] Eggert, D. and Bowyer, K. "Computing the Perspective Projection Aspect Graph of Solids of Revolution", *IEEE Trans. PAMI*, to appear, 1992.

[11] Eggert, D. "Aspect Graphs of Solids of Revolution", Doctoral Dissertation, Dept. of Computer Science & Engineering, University of South Florida, 1991.

[12] Eggert, D., Bowyer, K., Dyer, C., Christensen, H. and Goldgof, D. "The Scale Space Aspect Graph", Tech. Report, Dept. of Computer Science & Engineering, University of South Florida, 1991.

[13] Faugeras,O., Mundy,J., Ahuja,N., Dyer,C., Pentland,A., Jain,R., Ikeuchi,K. and Bowyer,K. "Panel theme: Why aspect graphs are not (yet) practical for computer vision", *IEEE Work. Directions in Automated CAD-Based Vision*, 98-104, 1991.

[14] Gigus, Z., Canny, J. and Seidel, R. "Efficiently computing and representing aspect graphs of polyhedral objects", *IEEE Trans. PAMI*, **13**, 542-551, 1991.

[15] Gualtieri, J.A., Baugher, S. and Werman, M. "The visual potential: One convex polygon", *CVGIP*, 46, 96-130, 1989.

[16] Kender, J.R. and Freudenstein, D.G. "What is a 'degenerate' view?", *ARPA IUW*, 589-598, 1987.

[17] Koenderink, J.J. "The structure of images", *Biol. Cybernetics*, 50, 363-370, 1984.

[18] Koenderink, J.J. Solid Shape, (MIT Press, Cambridge, Mass.), 1990.

[19] Koenderink, J.J. and van Doorn, A.J. "The internal representation of solid shape with respect to vision", *Biol. Cybernetics*, 32, 211-216, 1979.

[20] Kriegman, D. and Ponce, J. "Computing exact aspect graphs of curved objects: Solids of revolution", *IJCV*, 5, 119-135, 1990.

[21] Lindeberg, T. and Eklundh, J. "Scale detection and region extraction from a scale-space primal sketch", *ICCV*, 416-426, 1990.

[22] Malik, J. "Interpreting line drawings of curved objects", *IJCV*, 1, 73-103, 1987.

[23] Mokhtarian, F. and Mackworth, A.K. "Scale-based description and recognition of planar curves and two-dimensional shapes", *IEEE Trans. PAMI*, 8, 34-43, 1986.

[24] Mokhtarian, F. "Multi-scale description of space curves and three-dimensional objects", *CVPR*, 298-303, 1988.

[25] Plantinga, H. and Dyer, C.R. "Visibility, occlusion and the aspect graph", *IJCV*, 5, 137-160, 1990.

[26] Ponce, J. and Kriegman, D. "Computing exact aspect graphs of curved objects: Parametric surfaces", *Nat. Conf. Artificial Intelligence*, 1074-1079, 1990.

[27] Rieger, J. "The geometry of view space of opaque objects bounded by smooth surfaces", *Artificial Intelligence*, 44, 1-40, 1990.

[28] Sallam, M.Y., Stewman, J.S. and Bowyer, K.W. "Computing the visual potential of articulated assemblies", *ICCV*, 636-643, 1990.

[29] Seales,W.B. and Dyer,C.R. "Modeling the Rim Appearance", *ICCV*, 698-701, 1990.

[30] Sripradisvarakul, T. and Jain, R. "Generating aspect graphs for curved objects", *IEEE Work. Interpretation of 3D Scenes*, 109-115, 1989.

[31] Stewman, J.H. and Bowyer, K.W. "Creating the perspective projection aspect graph of convex polyhedra", *ICCV*, 494-500, 1988.

[32] Waldon, S., and Dyer, C. "Towards an Aspect Graph for Photometric Imaging". Tech. Report, Computer Science Dept., University of Wisconsin, 1992.

[33] Wang, R. and Freeman, H. "Object recognition based on characteristic view classes", *ICPR*, 8-12, 1990.

[34] Watts, N. "Calculating the principal views of a polyhedron", *ICPR*, 316-322, 1988.

[35] Witkin, A.P. "Scale-space filtering", in From Pixels to Predicates, (Ablex Pub. Corp., Norwood, NJ), 5-19, 1986.

[36] Yuille A. and Poggio, T. "Scaling theorems for zero crossings", *IEEE Trans. PAMI*, 8, 15-25, 1986.

# Achieving Generalized Object Recognition through Reasoning about Association of Function to Structure

Louise Stark and Kevin Bowyer

*Abstract*—Much current work in computer vision assumes that the recognition system will have a knowledge base consisting of, or derived from, an exact geometric model of each object that may be encountered. The purpose of the work described here is to demonstrate the feasibility of defining an object category in terms of the functional properties shared by all objects in the category. This form of representation should allow much greater generality. A complete system has been implemented that takes the boundary surface description of a 3-D object as its input and attempts to recognize whether the object belongs to the category chair and, if so, into which subcategory it falls. This is, to our knowledge, the first (only) implemented system to explore the use of a purely function-based definition of an object category (that is, no explicit geometric or structural model) to recognize 3-D objects. System competence has been evaluated on a database of over 100 objects, and the results largely agree with human interpretation of the objects.

*Index Terms*— Computer vision, function-based modeling, function-based object recognition, shape analysis, 3-D object representation.

Manuscript received October 1, 1990; revised December 30, 1990.

This work was supported by Air Force Office of Scientific Research grant AFOSR-89-0036, National Science Foundation grant IRI-8817776, and a Patricia Harris Fellowship.

The auhors are with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620.

IEEE Log Number 9102649.

## I. Introduction

Model-based vision has been popular for some time yet still appears far from being able to demonstrate any general-purpose 3-D object recognition system. One current "hot" paradigm is "CAD-based vision" — the use of exact geometric descriptions as might be available from a CAD system. With a CAD-based vision system, a unique 3-D model is stored for each object that the system is able to recognize. Recognition may require, in the worst case, that the input stimuli be compared to each model. Another problem encountered with such systems is that the size of the database grows in direct proportion to the number of objects the system is made capable of recognizing. One way of alleviating this problem to some degree is to allow parameterized representations so that objects that have the same essential geometry or structure can be recognized [2], [6], [7]. Still, it seems impossible to anticipate and parameterize all possible geometric and/or structural variations that may occur within an object category.

Consider the domain of human artifacts, that is, man-made objects that serve some specific purpose that is reflected in their external physical structure (e.g., furniture, hand tools, utensils). For any particular object category, there is some set of functional properties shared by all objects in that category. It is part of the thesis of our work that the existence or nonexistence of these properties can be deduced by analyzing the shape of an object and that this information can be used for recognition (or, if you like, categorization). Rather than concentrating our initial efforts on a purely theoretical elaboration of this concept, we have chosen to develop a complete system for a particular case study category. Our system represents the definition of object categories and subcategories in terms of required functional properties and represents the functional properties using procedural knowledge. A major advantage of this representation scheme is that the system can recognize truly novel objects, at least at the category level, even though *the system knows no specific geometric or structural model for any object.*

Section II reviews related research dealing with function-based representation. Section III describes the recognition system, followed by a detailed example and experimental results of the analysis of over 100 objects in Section IV. The paper concludes in Section V with suggestions for future directions of research.

Before proceeding, it is best to explicitly define some of the terminology we have adopted:

- *Category*: Using Rosch's terminology, we are considering the basic level category [10]. Rosch states that "basic categories are those which carry the most information, possess the highest category of validity, and are, thus, the most differentiated from one another" (see p. 382 of [10]).
- *Subcategory*: the term given subordinate categories (categories below the basic level). Each subcategory has its own set of functional attributes that may overlap with other subcategories.
- *Input Object*: an input to the system in the form of an uninterpreted 3-D boundary description.
- *Exemplar*: an object categorized by the system as belonging to a specific subcategory.
- *Functional Plan*: the function-based definition of a specific category or subcategory.
- *Function Label*: simply a name for the functional property being evaluated, for example, *provides sittable surface*.
- *Functional Element*: a portion of the input object that fulfills the functional requirements associated with a specific *function label*. There are three types of functional elements that can be identified: 1) a single surface of the object, such as the seat of a chair that provides a sittable surface; 2) a group of surfaces acting together to fulfill the required function, such as slats on the back of a chair act together to provide back support; 3) a three-dimensional portion (module) of the structure.
- *Association Measure*: a measure that reflects the strength of the association of the function label to the functional element or, cumulatively, the strength of the (sub)category membership of an object.
- *Procedural Knowledge Primitive (PKP)*: primitive procedures used to qualitatively evaluate the shape of an input object.

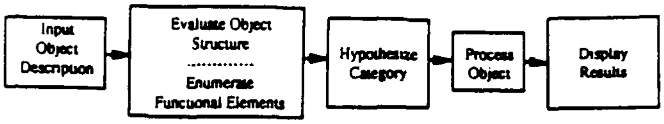


Fig. 1. Flow of execution.

## II. Background

Winston *et al.* have discussed the use of function-based definitions of object categories [13]. They point out that there can be an infinity of individual physical descriptions for objects in a category as simple as "cup" but that a single functional description can be used to represent all possible cups in a concise manner. This work is, of course, related to Winston's classic "arch-learning" program [14]. This earlier program was able to learn structural descriptions (*not* function-based descriptions) of object families, such as "arch," from line drawings of examples.

Brady *et al.* also discussed the relation between geometric structure and functional significance in their design of the "Mechanic's Mate" system [1], [3]. In part of this work, semantic net descriptions are computed from 2-D shapes, and a generalized structural description is learned from a sequence of positive examples.

Part of the inspiration for our work came from ideas expressed by Minsky in his recent book [9] and in network news articles. In fact, the category chair is used as an example by Minsky in his suggestion that knowledge about function must be combined with knowledge about structure.

Efforts that are more recent and closely related to ours are those of Ho [8] and of DiManzo *et al.* [4]. Ho considers two specific functional concepts (chair and support) in the context of what is needed to represent function for recognition. The analysis is done in the ideal 2-D cross section of the object and assumes that the object appears in its upright orientation. DiManzo proposes a system design that utilizes functional knowledge within an expert system framework. Primitives are defined in the form of individual expert systems that evaluate the 3-D information. A prototype system is being implemented that receives a description of a scene generated by an octree solid modeler.

## III. System Description

A high-level diagram of the system is depicted in Fig. 1.

This system reads the boundary description of an unknown 3-D polyhedral object in terms of face lists and vertex coordinates and, without user intervention, attempts to recognize whether the object belongs to the category chair and, if so, into which subcategory it falls. The size of the input object is treated as actual metric units so that objects may be "too big" or "too small" to function properly. (The system has the option of scaling the input object prior to analysis. The scale factor is calculated as the ratio of the volume of the convex hull of the input object to the volume of the convex hull of a "typical" straight back chair.)

In the first stage of the evaluation process, the input object is analyzed to identify all potential *functional elements*. This includes a list of individual surfaces (related to the faces of the object) and a list of combined surfaces. A *function label* can be associated to any of the three types of functional elements described above. The categorization performed by the system identifies functional elements of an input object by associating them with their proper function label.

At this time, the hypothesis of category chair is always made by the system without using any information derived from the structure of the input object. When the number of categories represented is expanded, heuristics will be invoked to hypothesize and prioritize a subset of categories. For example, one possible heuristic could evaluate the size of the object and select possible categories according to expected size ranges. For example, the 3-D volume of a couch would typically be much greater than a chair.

Processing of the object is guided by the function-based definition of the hypothesized category. This control structure holds the definition of the individual functional plans. Each functional plan has associated requirements. In turn, each requirement is processed as an ordered execution of primitives that qualitatively evaluate the input shape. We have identified a set of five PKP's that can be used to define functional requirements for the category *chair*.

The output of the system consists of whether the input object belongs to the category chair and, if so, into which subcategory it falls, as well as a cumulative association measure.

### A. Procedural Knowledge Primitives

Each function label is defined using a combination of PKP's. The PKP's currently used are *relative orientation*, *dimensions*, *stability*, *proximity*, and *clearance*. (This list is not assumed to be complete for all possible categories, but we expect it to be sufficient for the superordinate category furniture.) These primitives are procedures that make qualitative decisions about whether an object possesses a certain primitive property. During the initial system design, we began with a somewhat lengthier list of what we felt intuitively were the primitive functional concepts. As our system progressed, we often found that several of our intuitive primitives (for example, essentially parallel and essentially orthogonal) could be subsumed into one general routine (relative orientation), which was actually more useful (when we added the functional plan of the subcategory lounge chair).

The PKP *relative orientation* analyzes the orientation between two surfaces by evaluating the angle between the surface normals. For example, the sittable surface of the chair is expected to be essentially parallel to the ground plane in the chair's stable orientation. Some allowable ranges of orientation are more lenient than others. For example, the back support of a lounge chair can take on a large range of orientations relative to the sittable surface.

The PKP *dimensions* tests the potential functional element using multiple metrics. For example, the sittable surface of the chair is expected to be within a certain size range (depth and width) and to be situated within a set range above the ground (height).

The PKP *stability* is required for all subcategories of chair. For the sittable surface or seat rest to be maintained in its required orientation, the chair must *provide stable support*. Stable support is established by finding the convex hull of the contact points of the object with the ground plane in a given orientation. If a vector from the center of mass of the object perpendicular to the ground plane projects within the convex hull of the contact points, then the object is considered to be stable. To test if the object can act as a chair, the system applies weight to a distribution of points on the candidate sittable surface. This simply shifts the center of mass of the object, and therefore, the same stability test can be reapplied.

The *proximity* PKP tests to make sure two surfaces are in the proper proximity. For example, for a functional element to act as a back support, it must be close to the sittable surface and opposite an accessible area (i.e., the front of the seat). The surface must also be above the level of the sittable surface and be approximately centered relative to the sittable surface.

The PKP *clearance* is simple but extremely important. The functional elements may all be of the proper dimensions and be situated in the proper orientation to perform the functional requirements, but if the elements are not accessible by the user, they cannot be considered valid. Clearance is established by specifying the area that is expected to be accessible by the user and making sure there are no obstructions present. For example, the sittable surface must be clear above and "in front of" so that there is room for the person's torso and legs.

PKP's are invoked in a sequence dependent on the subcategory functional plan. All PKP's return an association measure that reflects how well the functional requirements are met.

### B. Structure of the Class Definition for Chair

The functional representation of each category is organized in a hierarchical graph (Fig. 2). This graph is also a control structure for the evaluation process. Each node of the graph is represented by a frame having four fields: *Name*, *Type*, *Realized By*, and *Functional Plans*. The Name field holds a unique identifier. Nodes are one of three types: *Category*, *Subcategory*, or *Function*. The root node in Fig. 2 is of type *Category*, being a basic-level category. The Functional Plans field has as many arcs as there are subcategories defined for that node. For example, in our current implementation, we have defined four subcategories: Conventional Chair, Balans Chair, Lounge Chair, and Highchair.

The graph structure of Fig. 2 represents our function-based description of the category *Chair*. Each subgraph formed with a subcategory frame as its root denotes a separate functional plan. Therefore, the function-based description of the subcategory Lounge Chair is realized by a totally different functional plan than that of the Balans Chair.

The final field of the frame is the *Realized By* field. This field points to an ordered list of function labels. The applicability of a given function label is evaluated by the sequence of PKP invocations associated with the function label node. For example, Conventional Chair requires the functions *provides sittable surface* and *provides stable support*. Both of these function labels must be satisfied at some threshold association measure in order to consider the object to be falling within the subcategory of Conventional Chair. It should be noted that there may be multiple potential results for a given object, each with its own association measure.

Each function label has its own specified constraint values for each PKP invocation depending on the functional requirement being evaluated. These values are stored in a constraint list that is associated to the category definition. The constraint list is made up of unique constraint identifiers, along with minimum, maximum, and average values for each. These constraint values have been gathered from sources that summarize the results of ergonomic design research [5].

The base values for the accumulation of the association measure originate with the PKP invocations. For a given PKP invocation, a qualitative decision is first made as to whether there is any functional element of the input object that satisfies the specified constraint range. If not, then a measure of zero is returned for the PKP invocation; otherwise, a list of functional elements with measures between zero and one is returned. This list of elements may then be input to another PKP invocation. If a required function label for a given (sub)category has no possible elements, then the association measure for the (sub)category may go to zero and further analysis for that (sub)category discontinued. The association measure is passed back to the current (sub)category, and the association measures of the

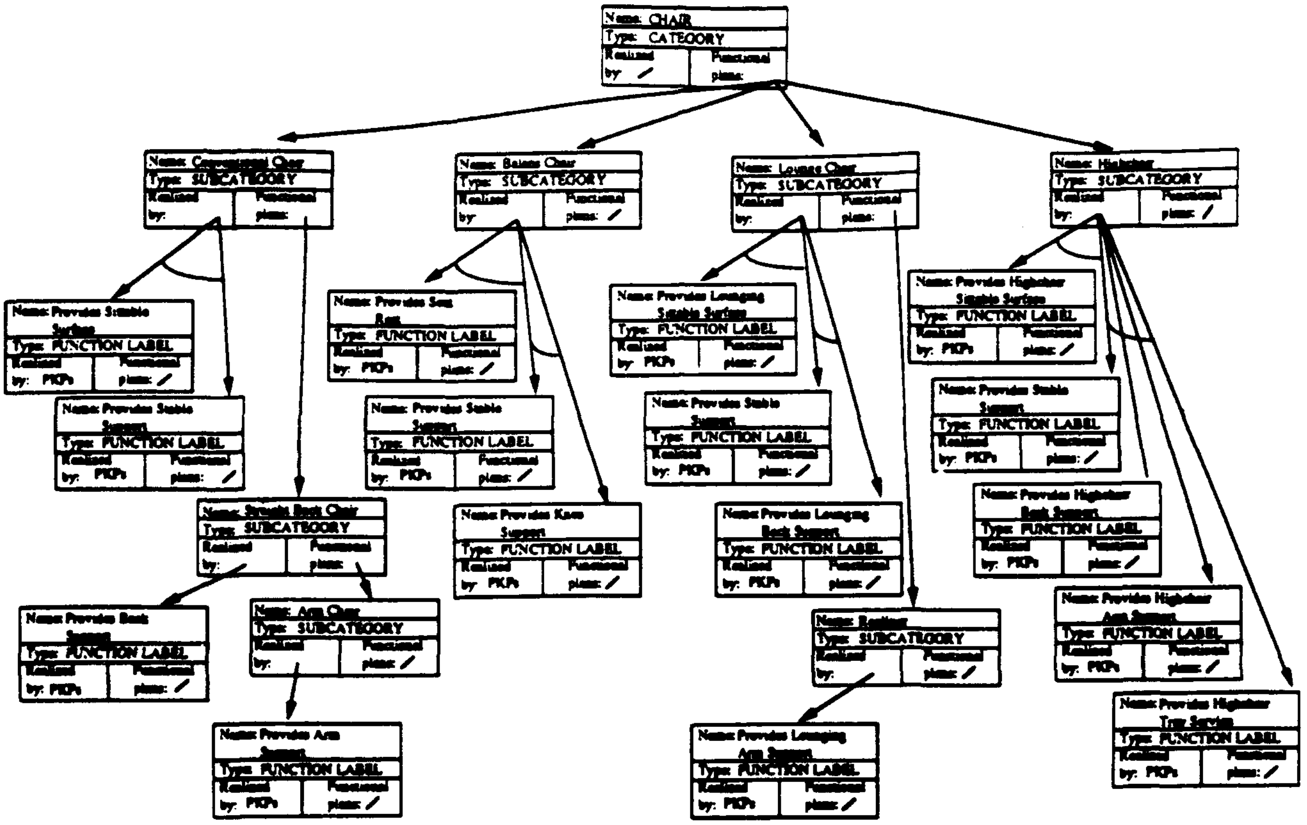


Fig. 2. Category representation graph.

different function labels are combined to determine the cumulative association measure for the (sub)category (see [12] for more details).

The category representation graph is the control structure for input object analysis. As the graph is traversed in depth-first fashion, if the (sub)category node has associated functional requirements, then those requirements are evaluated. If it is found that the requirements can be met by some portion(s) of the structure within some threshold association measure, then the functional elements are formed into a list. When applicable, the proper orientation for the object is also saved in the list.

The subcategory nodes are constrained by the information acquired from the parent subcategory nodes. This restriction is called *structural constraint propagation*. Many functional elements have an implied association that will constrain their possible structure and position. For example, the functional element that acts as the back of a chair for the subcategory Straight Back Chair must be situated above and approximately perpendicular to the functional element, found at the Conventional Chair level, which acts as the sittable surface.

If more than one function label is associated with a single (sub)category node, then the function label nodes are evaluated in a left to right manner. Therefore, referencing the functional requirements defined for the Conventional Chair (Fig. 2), the function label *provides sittable surface* must be fulfilled before initiating the procedural knowledge associated with *provides stable support*. This implies that *structural constraint propagation* exists between sibling function labels as well as between subcategory function labels.

## IV. Implementation

The system is implemented in C on a Sun workstation. Over 100 test objects, defined by a number of different individuals, have been analyzed. Each object definition is composed of a face file and a vertex file.[1] The recognition system reads each of these files along with the category definition file. This file holds the information in a format that can be read to construct the category representation graph.

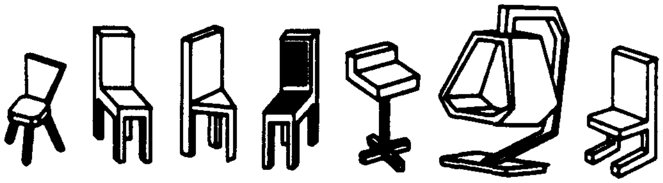


*Fig. 3. Example objects recognized as straight back chairs.*

The extent of how "generic" the function-based representation scheme actually is can best be seen in a sample of the objects that the system was capable of correctly categorizing. All of the objects appearing in Fig. 3 (along with many others) were categorized as straight back chairs.

Each fulfills the functional requirements of *provide sittable surface*, *provide stable support* and *provide back support* in its own way. In order to gain a better understanding of the reasoning process, a trace of the analysis of a simple example is now given. Fig. 4 depicts the input of an Arm Chair and the labeled output produced by the system.

The ground plane is considered to be parallel to the $X$-$Y$ plane. It is also assumed that gravity acts in the $-Z$ direction. As seen in Fig. 4, input objects do not have to be in "upright" orientation. The system's first step is to evaluate the shape of the input object. This consists of enumerating the surfaces and modules that can act as functional

[1] The collection of object descriptions used is available to interested researchers through anonymous ftp on figment.csee.usf.edu under *pub/errors_stuff/Objects*.

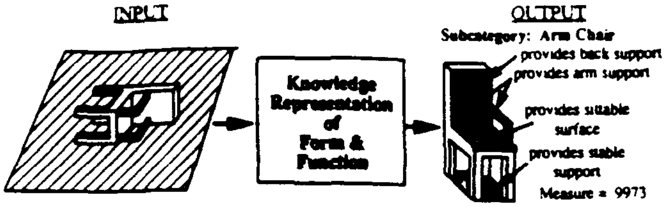


Fig. 4. Example input and output of system evaluation.

elements. Individual surfaces are listed, along with all surfaces that can be formed by grouping *essentially coplanar* surfaces. The object is further evaluated by subdivision into a set of convex 3-D modules, which are found directly from the object geometry. The center of mass of the whole object is calculated, along with the area of each of the surface functional elements.

### A. *Evaluation of 3-D Shape*

Evaluation begins with the category associated with the root node. Since there are no function labels associated with the node *Chair*, processing passes to the first *Subcategory Conventional Chair*. The list of PKP's invoked to realize the first function label *provides sittable surface* is shown in Fig. 5(a). The *dimensions* PKP finds all functional elements of the input object that are of the proper size range to be a sittable surface. This ensures that the "seat" of the chair is large enough to support the seat of a normal person and not so large that it could be a couch or table top. The surface, or group of surfaces, must also provide the proper amount of contiguous surface area. Surfaces that survive this test include what we would think of as the back of the chair, the seat of the chair, and the bottom of the chair.

The list of potential sittable surfaces found in the first procedure is passed to the next PKP *relative orientation*. This procedure attempts to confirm that the potential sittable surface is essentially parallel to the ground plane. If it is not, a transformation that will orient the potential sittable surface parallel to the ground plane is calculated and stored with the surface.

The next PKP uses information from the prior PKP's to test whether each potential sittable surface, when positioned parallel to the ground, can be within the proper height range. The potential sittable surface has been transformed such that the normal of the surface is aligned in the $+Z$ direction. The dimensions test finds the greatest distance spanned by the object in the $-Z$ direction. This gives a tentative height for the potential sittable surface. The back is eliminated in this test because there is no structure that can support the back in the proper height range. Two surfaces remain as potential sittable surfaces: the seat and the bottom of the chair.

The tests performed to this point are computationally simple tests that are used to prune the list of possible functional elements. The next two tests ensure that the surviving surfaces are clear and accessible for use.

A list of possible seat surfaces has now been identified (see Fig. 5(b)). If the list were empty, then it would be decided at this point that the object in question is not a conventional chair. An association measure of zero would be returned, and processing would continue with the next subcategory node Balans Chair. The association measure for each functional element found to this point is a function of the area and the potential height. Since the list is not empty, a list of potential sittable surfaces has been accumulated. This completes the tests associated with the procedural knowledge of *provides sittable surface*. The list of potential sittable surfaces is passed to the next function label node.

The second function to confirm is that the object has a base structure that *provides stable support*. The only PKP associated to this function label is *stability*. The procedure tests each potential result in its specified orientation. The object must be able to be placed in a stable position and still maintain the sittable surface in its proper orientation. To test for stability, each potential sittable surface is oriented in the $X$-$Y$ plane with the surface normal in the $+Z$ direction. The maximum $-Z$ displacement is found, and all vertices at this level are accumulated. These are potential points of contact with the ground to give support to the object. One of three conditions must exist: 1) Only a single point is in contact; 2) multiple collinear points are in contact; 3) at least three noncollinear points are in contact. In order to have sufficient contact, there must be at least three noncollinear points. Hence, if one of the first two conditions is found, then the object must be rotated such that at least three noncollinear points are in contact. This can lead to multiple possible new orientations to test. For each possible orientation, a list of contact points is accumulated. The convex hull of these points is then calculated to be used in the test for stability. It is assumed that the object has homogeneous density. Therefore, the force exerted downward can be represented with a single vector from the center of mass of the object pointing in the $-Z$ direction. If the force vector projects into the ground plane within the convex hull of the contact points, then the object is "self-stable." It is only considered "self-stable" because a force applied by the weight of a person does not have to be exerted directly over the center of mass of the object. This force can be applied in different positions downward on the sittable surface and tested to make sure that each resultant force (object plus applied weight) projects inside the convex hull.

Evidence is accumulated at the Conventional Chair node in support of the current hypothesis. The only surviving surface is, in fact, the seat of the chair (Fig. 5(d)). Face #20 (the bottom of the seat) was eliminated because stable support could not be verified.

The parsing of the object continues by checking the Straight Back Chair's associated function label. The list of PKP's used to confirm *provides back support* is given in Fig. 5(e). Each surface or group of surfaces that is *essentially orthogonal* to the potential sittable surface is tested. The *proximity* test checks to make sure the surface is close to and centered relative to the sittable surface. Clearance is also tested for the proposed back support relative to the potential sittable surface. There is only one surviving orientation at this point that provides all specified functions (Fig. 5(f)). This result is passed to the Arm Chair subcategory.

The list of PKP's used to realize *provides arm support* is depicted in Fig. 5(g). For a surface to act as an arm support, it must be oriented essentially parallel to the sittable surface. The arm support surfaces must be close and at the sides of the sittable surface. The surface must also be clear above for accessibility. One pair is found: one surface on each side of the sittable surface. These functional elements are labeled, and a new association measure is calculated.

Since there are no subcategories left in this subgraph, processing continues at the subcategory node Balans Chair. An association measure of zero is returned because the functional requirements of *provides seat rest* and *provides knee support* cannot be fulfilled by the structure of the arm chair. Association measures of zero are also found for the subcategory Lounge Chair and the subcategory Highchair, though for different reasons.

### B. *Experimental Results*

Each of the 101 input objects was designated as either CHAIR or NONCHAIR (see Figs. 6 and 7), based on the intuitive feelings of the designer. The objective was to compare the system's categorization to the intuitive categorization assigned by the designers. Table I summarizes the number of objects evaluated, the number categorized as CHAIR/NON-CHAIR by the designer, and corresponding numbers

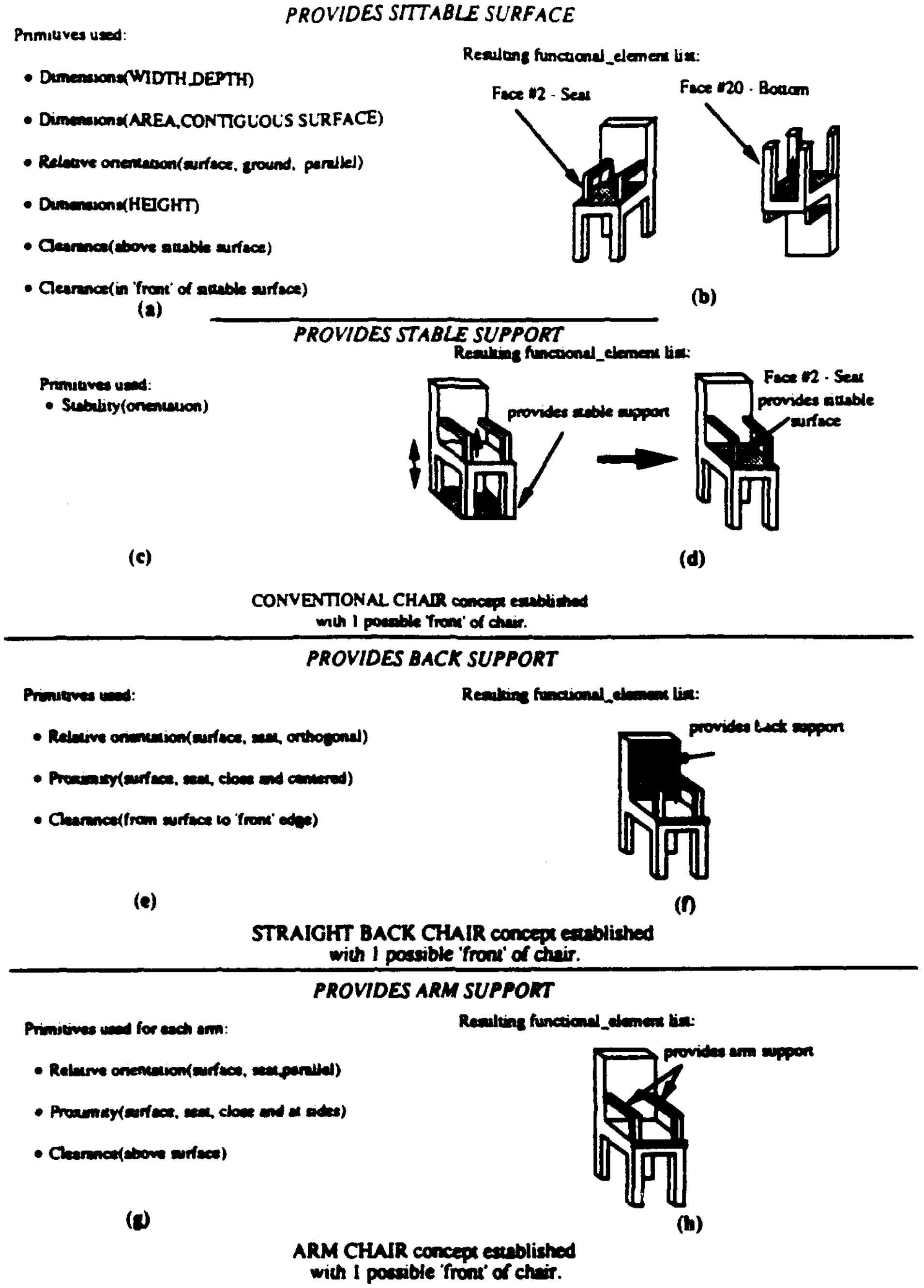

PROVIDES SITTABLE SURFACE
Primitives used:
• Dimensions(WIDTH,DEPTH)
• Dimensions(AREA,CONTIGUOUS SURFACE)
• Relative orientation(surface, ground, parallel)
• Dimensions(HEIGHT)
• Clearance(above sittable surface)
• Clearance(in 'front' of sittable surface)
(a)
Resulting functional_element list:
Face #2 - Seat
Face #20 - Bottom
(b)
PROVIDES STABLE SUPPORT
Primitives used:
• Stability(orientation)
Resulting functional_element list:
provides stable support
Face #2 - Seat
provides sittable surface
(c)
(d)
CONVENTIONAL CHAIR concept established
with 1 possible 'front' of chair.
PROVIDES BACK SUPPORT
Primitives used:
• Relative orientation(surface, seat, orthogonal)
• Proximity(surface, seat, close and centered)
• Clearance(from surface to 'front' edge)
Resulting functional_element list:
provides back support
(e)
(f)
STRAIGHT BACK CHAIR concept established
with 1 possible 'front' of chair.
PROVIDES ARM SUPPORT
Primitives used for each arm:
• Relative orientation(surface, seat,parallel)
• Proximity(surface, seat, close and at sides)
• Clearance(above surface)
Resulting functional_element list:
provides arm support
(g)
(h)
ARM CHAIR concept established
with 1 possible 'front' of chair.

Fig. 5. Evaluation process of arm chair.

for the system. There is only one input object intuitively categorized by its designer as a chair but not recognized as such by the system. This object (see Fig. 8(a)) was not categorized as a chair due to the fact that the system could not identify a contiguous sittable surface within the proper width/depth size range. The greatest discrepancy occurred with intuitively NONCHAIR objects that the system evaluated as being capable of functioning as a chair. Fig. 8(b) depicts all objects that were counter-intuitively identified by the system as falling into the Straight Back Chair subcategory. All of these objects have in common that they have some orientation in which they can provide a sittable surface, provide stable support, and provide a back support. They can all, therefore, *function* as Straight Back Chairs. Fig. 8(c) depicts the set of objects found to be capable of functioning as a Conventional Chair (i.e., provides sittable surface and provides stable support). One example of this is the *trash can* (object #2) in Fig. 8(c). By turning the trash can over, a person could use the bottom as a sittable surface.

## V. Future Research Directions

There are three areas we would like to investigate for extensions to the present system. First, the definition of more categories can be added to the knowledge base. We are completing the expansion of the system to include a number of basic level categories in the super-ordinate category "furniture." We also plan to add category representation from a different super-ordinate category, perhaps "dishes." This will allow us to test our assumption that the number

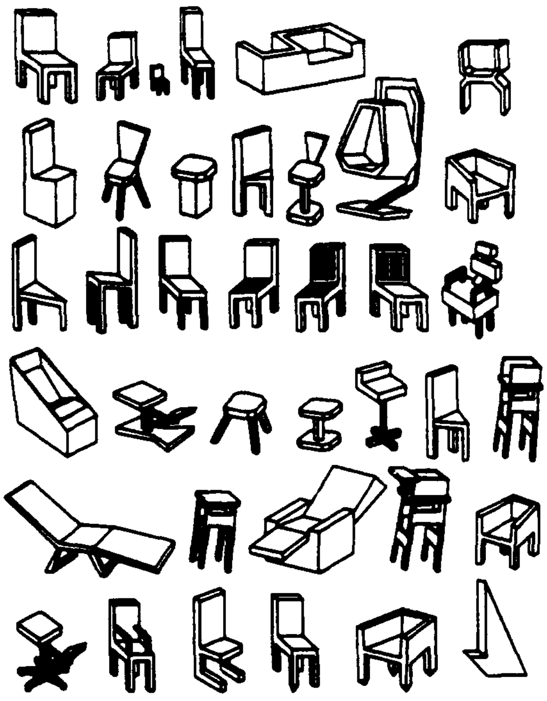


Fig. 6. Intuitive chair objects.

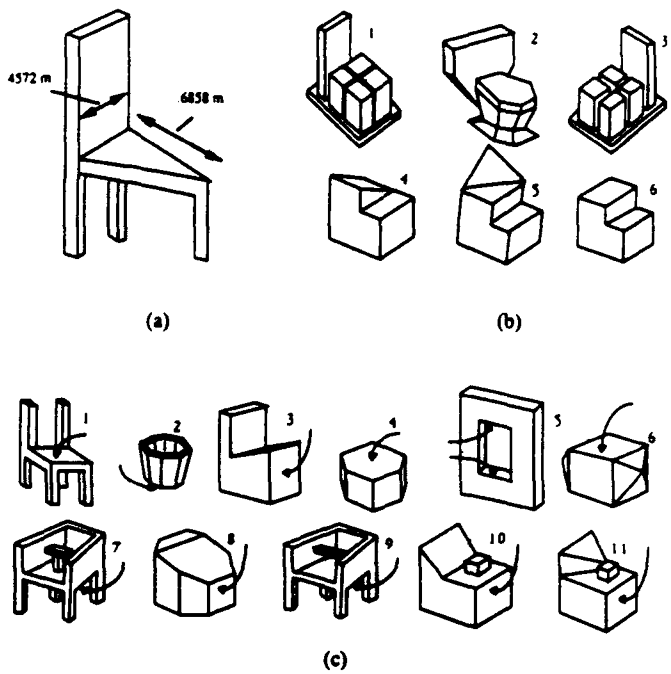


Fig. 8. Counter-intuitive chair results.

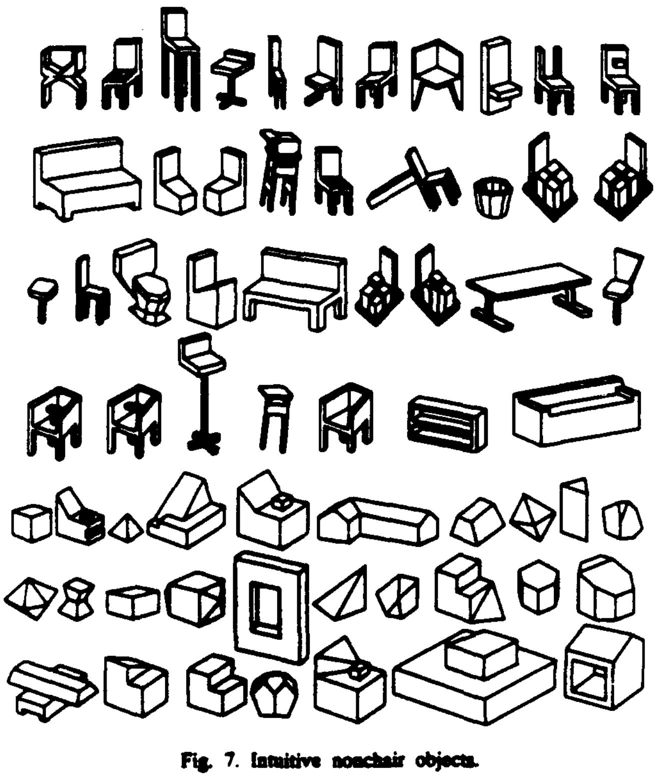


Fig. 7. Intuitive nonchair objects.

TABLE I
EXPERIMENTAL RESULTS

| | |
|---|---|
| Total # of objects | 101 |
| # Intuitively Categorized as Chairs | 38 |
| # categorized correctly as chairs by system | 37 |
| # Intuitively Categorized as Non-chairs | 63 |
| # categorized correctly as non-chairs by system | 46 |

of PKP's required grows very slowly with the number of categories. This will also allow us to investigate the formation of heuristics in hypothesizing the categories to use in the evaluation process. The structural information attained during the enumeration of functional elements could provide cues for the choice of hypothesis.

Second, we plan to investigate using nonideal input. Currently, the input objects examined by the system are "ideal" in that they are the output of a CAD tool. We hope to investigate the use of two forms of nonideal input. First, we want to explore the use of complete 3-D models constructed from multiple real images of an object. Second, we want to explore the use of incomplete 3-D models, as might be obtained from a single image and/or occluded views.

Third, we plan to investigate learning capabilities of the system. Through an interactive process, the system could question the user as to whether the structural differences found between objects categorized by the system have any functional significance. According to the user's response, new subcategories could be formed, and the control structure could be reorganized in such a way as to reflect the new functional plan. In this way, the system could learn by its experience.

## References

[1] M. Brady, P. E. Agre, D. J. Braunegg, and J. H. Connell, "The mechanics mate," in *Advances in Artificial Intelligence* (T. O'Shea, Ed.). New York: Elsevier, 1985, pp. 79–94.

[2] R. A. Brooks, and T. A. Binford, "Representing and reasoning about partially specified scenes," in *Proc. DARPA Image Understanding Workshop*, 1980, pp. 95–103.

[3] J. H. Connell and M. Brady, "Generating and generalizing models of visual objects," *Artificial Intell.* vol. 31, pp. 159–183, 1987.

[4] M. Di Manzo, E. Trucco, F. Giunchiglia, F. Ricci, "FUR: Understanding FUnctional Reasoning," *Int J. Intelligent Syst.*, vol. 4, pp. 431–457, 1989

[5] Eastman Kodak Co., *Ergonomic Design for People at Work.* New York: Van Nostrand Reinhold, 1983, vol. 1,

[6] R. R. Goldberg and D. G. Lowe, "Verification of 3-D parametric models in 2-D image data," in *Proc. IEEE Workshop Comput. Vision* (Miami FL), 1987, pp. 255–257.

[7] W. E. L. Grimson, "On the recognition of parameterized 2D objects," *Int. J. Comput. Vision*, vol. 3, pp. 353–372, 1988.

[8] S. Ho, "Representing and using functional definitions for visual recognition," Ph. D. Dissertation, Univ. Wisconsin, Madison, 1987.

[9] M. Minsky, *The Society of Mind*. New York: Simon and Shuster, 1985.

[10] E. Rosch, C. B. Mervis, W. D. Gray, D. Johnson, P. Boyes-Braem, "Basic objects in natural categories," *Cognitive Psych.*, vol. 8, pp. 382–439, 1976.

[11] L. Stark, and K. W. Bowyer, "Achieving generalized object recognition through reasoning about association of function to structure," *AAAI-90 Workshop Qualitative Vision* (Boston, MA), July 1990, pp. 137–141.

[12] L. Stark, L. O. Hall, and K. W. Bowyer, "An investigation of methods of combining functional evidence for 3-D object recognition," in *Proc. SPIE 1381: Intell. Robots Comput. Vision IX: Algorithms Techn.* (Boston, MA), 1990.

[13] P. H. Winston, T. O. Binford, B. Katz, and M. Lowry, "Learning physical description from functional definitions, examples, and precedents," in *Proc. Int. Symp. Robotics Research* (M. Brady and R. Paul, Eds.). Cambridge, MA: MIT Press, 1984, vol. 1.

[14] P. H. Winston, "Learning structural descriptions from examples," in *The Psychology of Computer Vision*, P. H. Winston (Ed.). New York: McGraw-Hill, 1975.